Map Cube: A Visualization Tool for Spatial Data Warehouses

S. Shekhar, C. Lu, X. Tan, S. Chawla, R. R. Vatsavai
Computer Science Department, University of Minnesota
200 Union Street SE, Minneapolis, MN-55455
[*shekhar, ctlu, xtan, chawla, vatsavai*]@cs.umn.edu TEL:(612) 6248307 FAX:(612)6250572
http://www.cs.umn.edu/Research/shashi-group

December 13, 1999

Abstract

Data warehouses(DW) are becoming essential tools in decision making and data analysis. Data cube operator is used to generate the union of a set of alpha-numeric summary tables corresponding to a given aggregation hierarchy. Spatial data warehouses prefer browsing of aggregated data in terms of albums of maps rather than the alpha-numeric summary tables. It is quite tedious to convert the output of data cube operator to an album of maps using current tools. We extend the concept of data cube to spatial domain by proposing "map cube," an operator which takes the base map, base table, cartographic preference, etc., and generates an album of maps. Map cube organizes the album of generated maps using the given aggregation hierarchy to support browsing via roll-up, drill-down, and other operators on aggregation hierarchy. We use the census data to illustrate the notion of the map cube, and discuss research issues raised by the map cube operator.

Keywords: Map Cube, Data Cube, Spatial Data Warehouse, Maps, GIS

1 Introduction

A data warehouse(DW) is a collection of decision support technologies, aimed at enabling knowledge workers(executives, managers, analysts) to make better and faster decisions [3]. Data warehouses contain large amounts of information, which is collected from a variety of independent sources and is often maintained separately from the operational databases. Traditionally operational databases are optimized for on-line transaction processing (OLTP), where consistency and recoverability are critical. Transactions are typically small, and access a small number of records based on the primary key. Operational databases maintain current state information. In contrast, data warehouses maintain historical information and are designed for on-line analytical processing (OLAP), where queries aggregate large volumes of data in order to detect trends and anomalies [17]. The academic research in DW addresses issues such as data warehouse design processes [12], data cube models [8] and efficient implementations [10, 11]. Dimensions, measures and aggregation hierarchies are the core concepts in data warehouses. Dimensions are used to define the context of the measures, and can be viewed as independent variables (or keys) which determine the values of measures (dependent variables). There are two kinds of aggregation hierarchies, namely, dimension power-set hierarchy and per-dimension concept hierarchy. To facilitate the complex analysis, the data in the warehouse are often modeled as a multi-dimensional data cube. For example, in a census data warehouse, the age group, income type, race category, and time (year) are some of the dimensions of interest. Data cube [8] is an aggregate operator, and it computes all group-by SQL queries and aggregates data.

Spatial data warehouses contain geographic data, e.g., satellite images, remote sensing images [5, 16, 11], etc., in addition to non-spatial data. Examples of spatial data-warehouses include the US Census data-set [1, 6], Earth Observation System archives of satellite imagery [22], Sequoia 2000 [20], highway traffic measurement archives etc. The research in spatial data warehouses has focused on case-studies [5, 16] and on per dimension concept hierarchy [11]. In contrast, in this paper, we will focus on the dimension power-set hierarchy. A major difference between conventional and spatial data warehouses lies in the visualization of the results. Conventional data warehouse OLAP results are often shown as summary tables or spread sheets of text and numbers, whereas in the case of spatial data warehouse the results may be albums of maps. It is not trivial to convert the alpha-numeric output of data cube on spatial data warehouses into an organized collections of maps. Another issue is the aggregate operators on geometry data types(e.g. point, line, polygon). Neither existing databases nor the emerging standard for geographic data, OGIS [18], has addressed this issue. In this paper we present *map cube*, an operator based on conventional data cube but extended for spatial data warehouses. With the *map cube* operator, we visualize the data cube in spatial domain via an album of maps. For some spatial applications, e.g. census data, which require aggregation on different measures and comparison between different categories in each attribute, the map cube operator can be very useful.

Map cube is an operator which takes a base map, associated data tables, aggregation hierarchy and cartographic preferences to produce an album of maps. This album of maps is organized using the given aggregation hierarchy(e.g. dimension cube). The goal is to support exploration of the map collection via roll-up, drill-down, and other operations on aggregation hierarchy. We also provide a set of aggregate operators for geometric data types and classify them using a well-known classification scheme for aggregate functions.

The rest of the paper is organized as follows. Section 2 discusses some basic concepts of Data Warehouses and Geographic Information System. In section 3, the definition and operation of map cube are introduced with an example. Section 4 shows an application of the map cube. In Section 5, the research issues related to map cube are discussed. Section 6 includes a summary and a discussion of future directions.

2 Basic Concepts

2.1 Concepts in Geographic Information System

A *geographic information system (GIS)* [4, 23] is a computer-based information system consisting of integrated set of programs and procedures which enable the capture, modeling, manipulation, retrieval, analysis and presentation of geographically referenced data. A map is a collection of vector and raster layers as shown in Figure 1. Each map has its own visual representation, including graphics, layout, legend, title, etc.

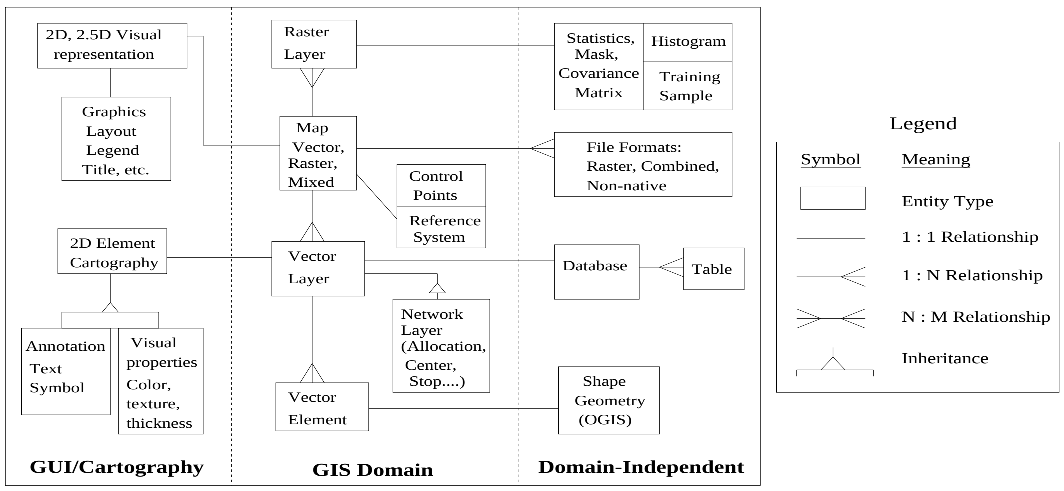


Figure 1: Concepts in Geographic Information Systems

In the raster layer representation, the space is divided into cells. The locations of a geographic object or conditions are defined by the row and column of the cells they occupy. The area that each cell represents defines the spatial resolution available. The raster layer is a collection of pixels and may represent raw images collected directly from satellites, aerial photography and other sensors. The raster layer may also represent interpreted images showing a classification of areas.

Information associated with the raster layer representation include statistics mask, covariance matrix, histogram for the classified images, and training sample. The training sample may be used by supervised classification. These training fields are areas of known identity delineated on the digital image, usually by specifying the corner points of a rectangular or polygonal area using line and column numbers within the coordinate system of the digital image.

Vector layers are collection of vector elements. The shape of vector element may be zero dimensional(point), one dimensional(curves, lines) or two dimensional(surface, polygons). For example, in a vector layer representation, an object type *house* may have attributes, referencing further objects types: polygon, person name, address, and date. The polygon for the house is stored as a "Vector Element" in Figure 1. A vector layer may have its own cartographic preferences to display the elements. These cartographic preferences include text, symbol, and some visual properties: color, texture, thickness, etc.

The elements and attributes in the vector layers may be associated with non-spatial attributes managed by a database system which consists of many tables. In Figure 4, for example, the vector layer Base-Map has its corresponding table, Election-Base, which has the attributes of State Name, GPP, LPP, Boundary, and Delegates. The Boundary is a foreign key pointing to another table which describes the geometric boundary of each polygon.

A network layer is a special case of a vector layer. The network layer is composed of a finite collection of the points, the line-segments connecting the points, the location of the points, and the attributes of the points and line-segments. For example, the network layer for transportation applications may store road intersection points and the road segments connecting the intersections.

Maps are also associated with reference systems and control points. A reference system is a coordinate system attached to the surface of the earth. Reference system allows us to locate the map element on the surface of the earth. Control points are common to the base map and the slave map being prepared. The exact locations of control points are well defined. Examples include intersection of roads and railways or other land marks or monuments. The control points are used to geo-register newly acquired maps to the well-defined base map at different scales.

2.2 Aggregate Functions

Aggregate functions compute statistics for a given set of values. Examples of aggregate functions include sum, average, and centroid. Aggregate functions can be grouped into three categories, namely, distributive, algebraic, and holistic as suggested in [8]. We define these functions in this section and provide some examples from GIS domain. Table 1 shows all of these aggregation functions for different data types.

	Aggregation Function		
Data Type	Distributive Function	Algebraic Function	Holistic Function
Set of numbers	Count, Min, Max, Sum	Average, Standard Deviation, MaxN, MinN()	Median, MostFrequent, Rank
Set of points, lines, polygons	Convex Hull, Geometric Union, Geometric Intersection	Centroid, Center of mass, Center of gravity	Nearest neighbor index, Equi-partition

Table 1: Aggregation Operations

- **Distributive**: An aggregate function F is called distributive if there exists a function G such that the value of F for an N-dimensional cube can be computed by applying G function to the value of F for $(N+1)$-dimensional sub-cubes. For N=1, $F(M_{ij}) = G(F(C_j)) = G(F(R_i))$, where M_{ij} represents the elements of 2-dimensional matrix, C_j denotes each column of the matrix, and R_i denotes each row of the matrix. Consider the aggregate function MIN and COUNT as shown in Figure 2. In the first example, $F = MIN$, then $G = MIN$, since $MIN(M_{ij}) = MIN(MIN(C_j)) = MIN(MIN(R_i))$. For the second example, $F = COUNT$, $G = SUM$, since $COUNT(M_{ij}) = SUM(COUNT(C_j)) = SUM(COUNT(R_i))$. Other distributive aggregate functions include MAX(), SUM(), etc. Note that "null" valued elements are ignored in computing aggregate functions.

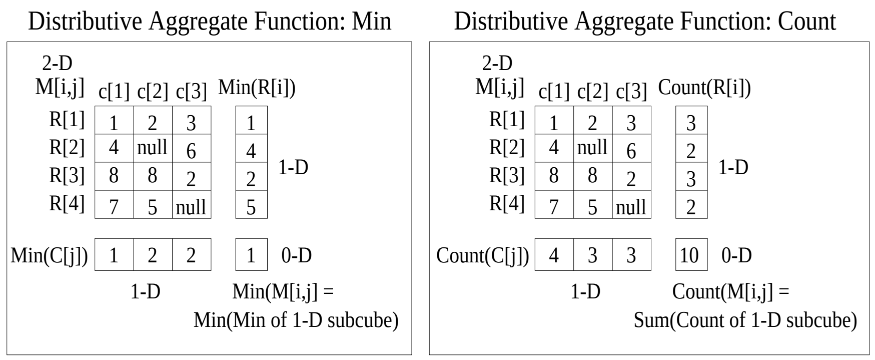


Figure 2: Computation of distributive aggregate function

  Distributive GIS aggregate operations include Convex hull, Geometric Union, Geometric Intersection, etc. The convex hull of a set Q of points is the smallest convex polygon P for which each point in Q is either on the boundary of P or in its interior. Intuitively, the convex hull is the shape formed by a tight rubber band that surrounds all the nails. The geometric intersection is a binary operation that takes two sets of geometric areas and return the set of regions that are covered by both of the original areas. The geometric union is a binary operation that takes two sets of geometric areas and return the set of regions that are covered by at least one of the original areas. For all of these aggregations, the operator aggregates the computed regions of the subset, and then computes the final result.

- **Algebraic**: An aggregate function F is algebraic if F of an N-dimensional cube can be computed using a fixed number of aggregates of the (N+1)-dimensional sub-cubes. Average, Variance, Standard Deviation, MaxN, MinN are all algebraic. In Figure 3, for example, the computations of Average and Variance for the matrix M are shown. The average of elements in two dimensional matrix M, can be computed from Sum and Count values of the 1-D sub-cubes(e.g. rows or columns). The Variance can be derived from, Count, Sum(i.e. $\sum_i X_i$), and Sum of Sq(i.e. $\sum_i X_i^2$), of rows or columns. Similar techniques apply to other algebraic functions.

  An algebraic aggregate operation in GIS is Center. The center of n geometric points $\vec{V}^i = (V_x^i, V_y^i)$ is defined as $C\hat{e}nter = \frac{1}{n}\sum \vec{V}_i$, $C_x = \frac{\sum V_x}{n}$, $C_y = \frac{\sum V_y}{n}$. Both the center and the count are required to compute the result for the next layer. Center of mass and

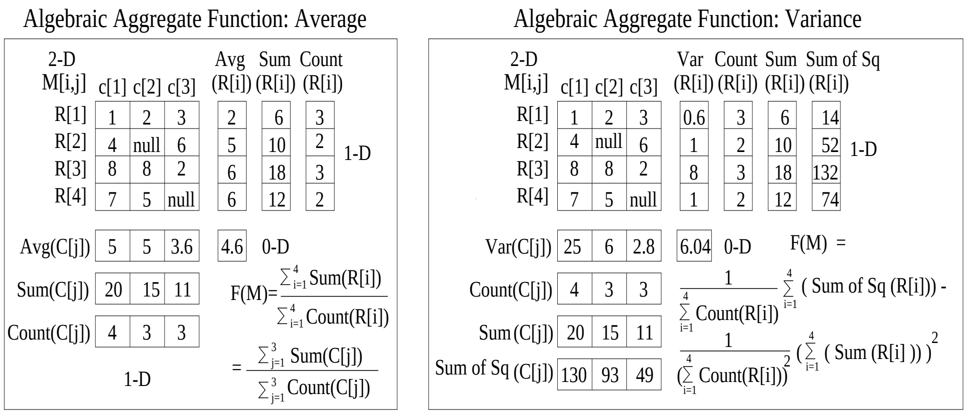

Algebraic Aggregate Function: Average
2-D M[i,j]
c[1] c[2] c[3]
Avg (R[i]) Sum (R[i]) Count (R[i])
R[1] 1 2 3 | 2 6 3
R[2] 4 null 6 | 5 10 2
R[3] 8 8 2 | 6 18 3
R[4] 7 5 null | 6 12 2
1-D
Avg(C[j]) 5 5 3.6 | 4.6 0-D
Sum(C[j]) 20 15 11
Count(C[j]) 4 3 3
1-D
F(M)= Σ_{i=1}^4 Sum(R[i]) / Σ_{i=1}^4 Count(R[i]) = Σ_{j=1}^3 Sum(C[j]) / Σ_{j=1}^3 Count(C[j])
Algebraic Aggregate Function: Variance
2-D M[i,j]
c[1] c[2] c[3]
Var (R[i]) Count (R[i]) Sum (R[i]) Sum of Sq (R[i])
R[1] 1 2 3 | 0.6 3 6 14
R[2] 4 null 6 | 1 2 10 52
R[3] 8 8 2 | 8 3 18 132
R[4] 7 5 null | 1 2 12 74
1-D
Var(C[j]) 25 6 2.8 | 6.04 0-D
Count(C[j]) 4 3 3
Sum (C[j]) 20 15 11
Sum of Sq (C[j]) 130 93 49
F(M) = 1 / Σ_{i=1}^4 Count(R[i]) Σ_{i=1}^4 (Sum of Sq (R[i])) - 1 / (Σ_{i=1}^4 Count(R[i]))^2 (Σ_{i=1}^4 (Sum (R[i])))^2

Figure 3: Computation of algebraic aggregate function

Center of gravity are other examples of algebraic aggregate functions.

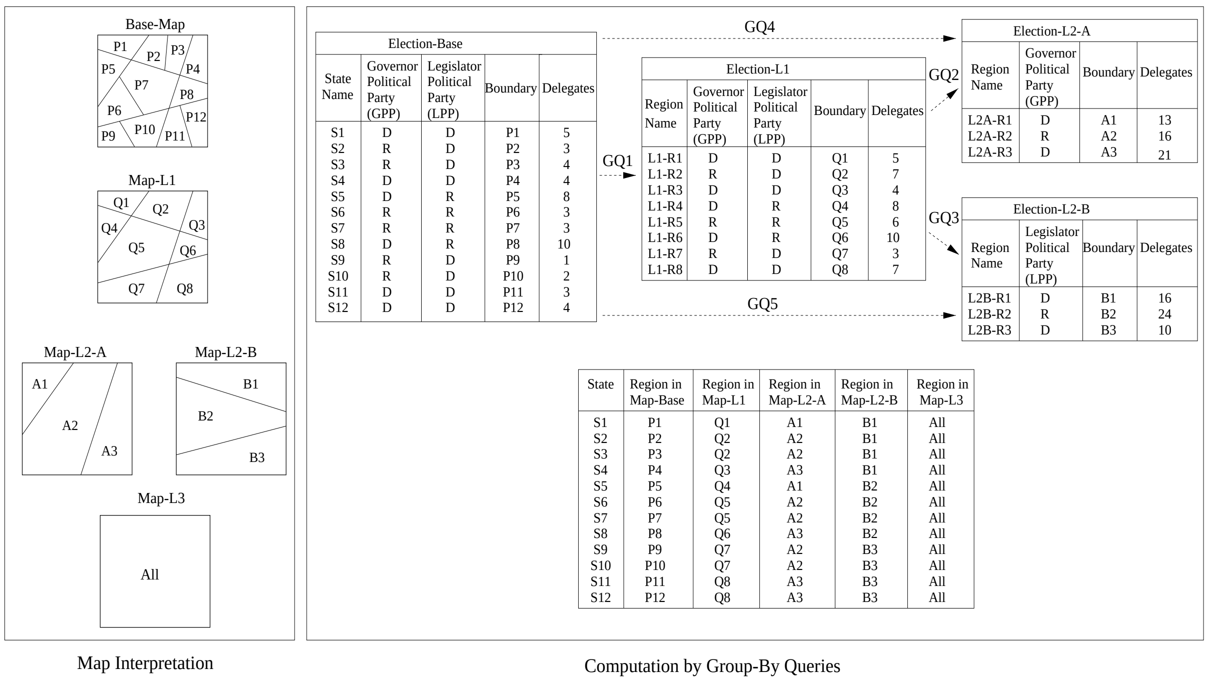
- **Holistic**: An aggregate function F is called holistic if the value of F for an N-dimensional cube can not be computed from a constant number of aggregates of the (N+1)-dimensional sub-cubes. To compute the value of F in each level, we need to access the base data. Examples of holistic function include Median, MostFrequent, and Rank.

  Holistic GIS aggregate operations include equi-partition and nearest-neighbor index. Equi-partition of a set of points yields a line L such that there are the same number of point objects on each side of L. Nearest-neighbor index measures the degree of clustering of objects in a spatial field. If a spatial field has the property that like values tend to cluster together, then the field exhibits high nearest-neighbor index. When new data are added, many of the tuples in the nearest neighbor relationship may change. Therefore, the nearest-neighbor index is holistic. The line of Equi-partition could be changed with any new added points. To compute the equi-partition or nearest neighbor-index in all levels of dimensions, we need the base data.

The computation of aggregate functions has graduated difficulty. The distributive function can be computed from the next lower level dimension values. The algebraic function can be computed from the next lower scratchpads. The holistic function needs the base data to compute the result in all levels of dimension.

An example of geometric aggregation

Figure 4 shows the results from aggregation operation of geometric union. The base table, Election-Base, has the following attributes: State Name, Governor Political Party(GPP), majority Legislator Political Party(LPP), Boundary, and Delegates. The Boundary is a foreign key pointing to another table which describes the geometric polygon representing the state boundary polygon. From the base table and its corresponding map, we issue the queries GQ1, GQ2, GQ3, GQ4, and GQ5 as listed in Table 2. If the neighboring polygons have the same value in the attributes of the GROUP BY clause, the Geometric-Union-by-Continuous-Polygon operator merges them into one large polygon. These queries have the same effect as the GIS



Election-Base				
State Name	Governor Political Party (GPP)	Legislator Political Party (LPP)	Boundary	Delegates
S1	D	D	P1	5
S2	R	D	P2	3
S3	R	D	P3	4
S4	D	D	P4	4
S5	D	R	P5	8
S6	R	R	P6	3
S7	R	R	P7	3
S8	D	R	P8	10
S9	R	D	P9	1
S10	R	D	P10	2
S11	D	D	P11	3
S12	D	D	P12	4

Election-L1				
Region Name	Governor Political Party (GPP)	Legislator Political Party (LPP)	Boundary	Delegates
L1-R1	D	D	Q1	5
L1-R2	R	D	Q2	7
L1-R3	D	D	Q3	4
L1-R4	D	R	Q4	8
L1-R5	R	R	Q5	6
L1-R6	D	R	Q6	10
L1-R7	R	D	Q7	3
L1-R8	D	D	Q8	7

Election-L2-A			
Region Name	Governor Political Party (GPP)	Boundary	Delegates
L2A-R1	D	A1	13
L2A-R2	R	A2	16
L2A-R3	D	A3	21

Election-L2-B			
Region Name	Legislator Political Party (LPP)	Boundary	Delegates
L2B-R1	D	B1	16
L2B-R2	R	B2	24
L2B-R3	D	B3	10

State	Region in Map-Base	Region in Map-L1	Region in Map-L2-A	Region in Map-L2-B	Region in Map-L3
S1	P1	Q1	A1	B1	All
S2	P2	Q2	A2	B1	All
S3	P3	Q2	A2	B1	All
S4	P4	Q3	A3	B1	All
S5	P5	Q4	A1	B2	All
S6	P6	Q5	A2	B2	All
S7	P7	Q5	A2	B2	All
S8	P8	Q6	A3	B2	All
S9	P9	Q7	A2	B3	All
S10	P10	Q7	A2	B3	All
S11	P11	Q8	A3	B3	All
S12	P12	Q8	A3	B3	All

Figure 4: An example of GIS aggregate function, geometric-union

GQ1	**SELECT** GPP, LPP, Geometric-Union-by-Continuous-Polygon(Boundary) **FROM** Election-Base **GROUP BY** GPP, LPP
GQ2	**SELECT** GPP, Geometric-Union-by-Continuous-Polygon(Boundary) **FROM** Election-L1 **GROUP BY** GPP
GQ3	**SELECT** LPP, Geometric-Union-by-Continuous-Polygon(Boundary) **FROM** Election-L1 **GROUP BY** LPP
GQ4	**SELECT** GPP, Geometric-Union-by-Continuous-Polygon(Boundary) **FROM** Election-Base **GROUP BY** GPP
GQ5	**SELECT** LPP, Geometric-Union-by-Continuous-Polygon(Boundary) **FROM** Election-Base **GROUP BY** LPP

Table 2: SQL queries for map reclassification

reclassify map operation. For example, the query GQ1 generates the same result as to reclassify Base-Map by attributes GPP and LPP. The map interpretation of these queries is shown in the left portion of Figure 4. The boundaries of regions $Q1, Q2, \ldots, Q8$ are derived from the geometric union of smaller regions $P1, P2, \ldots, P12$ in the Base-map. For example, $Q2$ represents the geometric union of $P2$ and $P3$ since they have same value for grouping attribute set <LPP,GPP>. Similarly, regions $A1, A2, A3$ in Map-L2-A and regions $B1, B2, B3$ in Map-L2-B are derived from the geometric-union of smaller regions in Map-L1(or Base-map). For example, region $A1$ in Map-L2-A is the geometric union of region $Q1$ and $Q4$ from Map-L1.

2.3 Aggregation hierarchy

The CUBE operator [8] generalizes the histograms, cross-tabulation, roll-up, drill-down, and sub-total constructs. It is the N-dimensional generalization of simple aggregate functions. Figure 5 [8] shows the concept for aggregation up to 3-dimensions. The dimensions are Year, Company, and Region. The measure is the sales. The 0D data cube is a point, which shows the total summary. There are three 1-D data cubes: Group-by Region, Group-by Company, and Group-by Year. The three 2-D data cubes are cross tabs, which are the combination of these three dimensions. The 3D data cube is a cube with three intersecting 2D cross tabs. Figure 6 [8] shows the tabular forms of total elements in the 3D data cube after the CUBE operation. Creating a data cube requires generating the power set of the aggregation columns.

A tabular view of the individual sub-space data-cubes of Figure 5 is shown in Figure 7. The union of all the tables in Figure 7 yields the resulting table from the data cube operator. The 0-dimensional sub-space cube labeled "Aggregate" in Figure 5 is represented by Table "SALES-L2" in Figure 7. The one-dimensional sub-space cube labeled "By Company" in Figure 5 is represented by Table "SALES-L1-C" in Figure 7. The two-dimensional cube labeled "By Company & Year" is represented by Table "SALES-L0-C" in Figure 7. Readers can establish correspondence between remaining subspace cubes and tables.

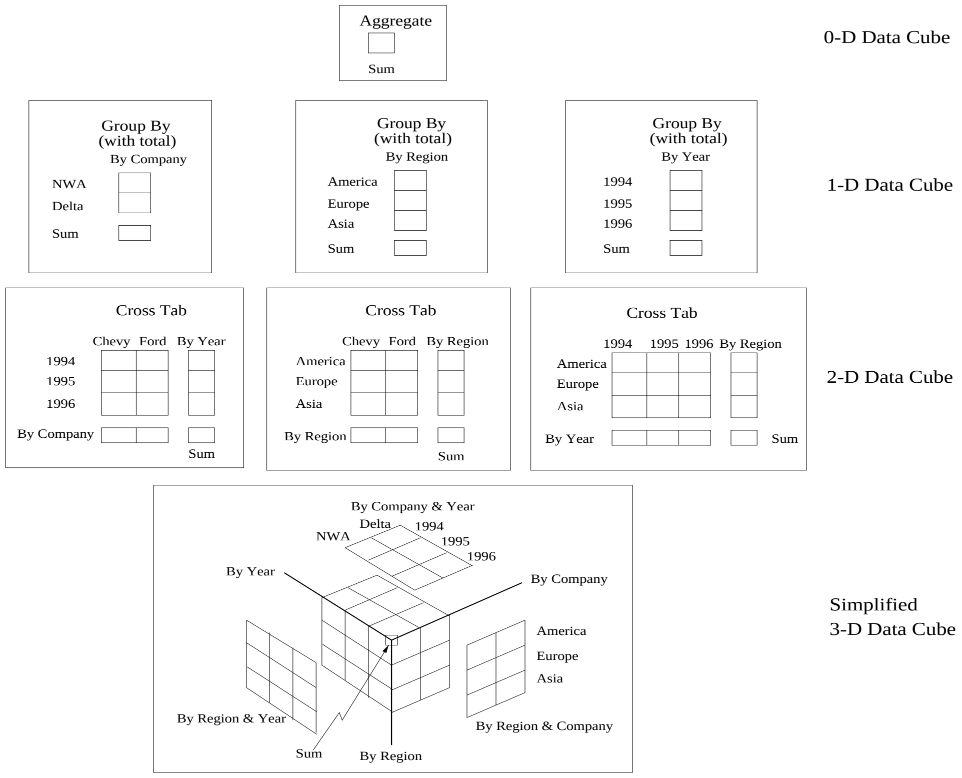
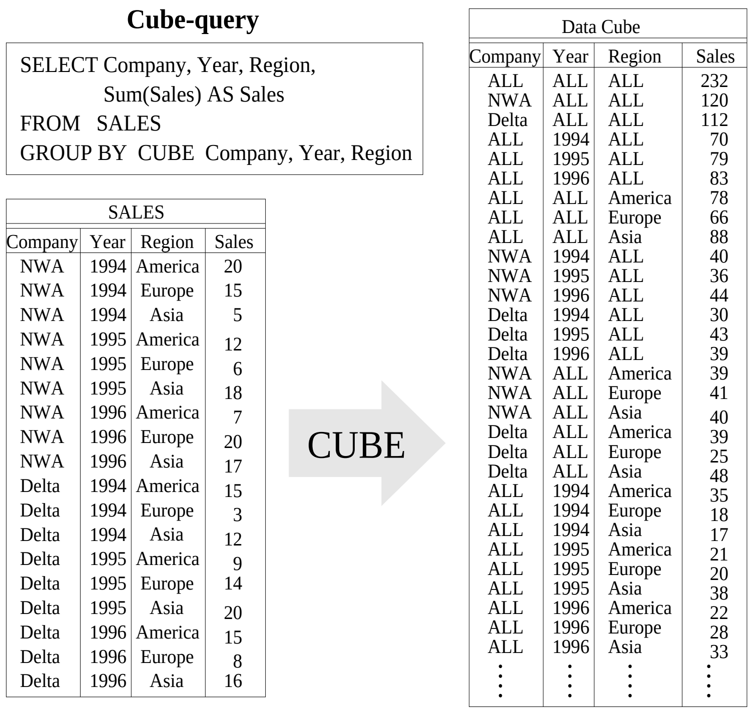


Figure 5: The 0-d, 1-D, 2-D, and 3-D data cubes

The cube operator can be modeled by a family of SQL queries using *GROUP BY* operators and aggregation functions. Each arrow in Figure 7 will be represented by a SQL query. In Table 3, we provide the corresponding queries for the five arrows labeled $Q1, Q2, \ldots, Q5$ in Figure 7. For example, query $Q1$ in Table 3 aggregates "Sales" by "Year" and "Region," and generates Table "SALES-L0-A" in Figure 7.

The *GROUP BY* clause specifies the grouping attributes, which should also appear in the


Cube-query

```
SELECT Company, Year, Region,
           Sum(Sales) AS Sales
FROM   SALES
GROUP BY  CUBE  Company, Year, Region
```

SALES

Company	Year	Region	Sales
NWA	1994	America	20
NWA	1994	Europe	15
NWA	1994	Asia	5
NWA	1995	America	12
NWA	1995	Europe	6
NWA	1995	Asia	18
NWA	1996	America	7
NWA	1996	Europe	20
NWA	1996	Asia	17
Delta	1994	America	15
Delta	1994	Europe	3
Delta	1994	Asia	12
Delta	1995	America	9
Delta	1995	Europe	14
Delta	1995	Asia	20
Delta	1996	America	15
Delta	1996	Europe	8
Delta	1996	Asia	16

Data Cube

Company	Year	Region	Sales
ALL	ALL	ALL	232
NWA	ALL	ALL	120
Delta	ALL	ALL	112
ALL	1994	ALL	70
ALL	1995	ALL	79
ALL	1996	ALL	83
ALL	ALL	America	78
ALL	ALL	Europe	66
ALL	ALL	Asia	88
NWA	1994	ALL	40
NWA	1995	ALL	36
NWA	1996	ALL	44
Delta	1994	ALL	30
Delta	1995	ALL	43
Delta	1996	ALL	39
NWA	ALL	America	39
NWA	ALL	Europe	41
NWA	ALL	Asia	40
Delta	ALL	America	39
Delta	ALL	Europe	25
Delta	ALL	Asia	48
ALL	1994	America	35
ALL	1994	Europe	18
ALL	1994	Asia	17
ALL	1995	America	21
ALL	1995	Europe	20
ALL	1995	Asia	38
ALL	1996	America	22
ALL	1996	Europe	28
ALL	1996	Asia	33
⋮	⋮	⋮	⋮

Base table

Resulting table from Cube operator
(aka data cube)

Figure 6: An example of data cube

SELECT clause, so that the value resulting from applying each function to a group of tuples appears along with the value of the grouping attribute(s).

Q1	**SELECT** 'ALL', Year, Region, **SUM**(Sales) **FROM** SALES-Base **GROUP BY** Year,Region
Q2	**SELECT** 'ALL', 'ALL', Region **SUM**(Sales) **FROM** SALES-L0-A **GROUP BY** Region
Q3	**SELECT** 'ALL', 'ALL', 'ALL' **SUM**(Sales) **FROM** SALES-L1-A
Q4	**SELECT** 'ALL', 'ALL', Region, **SUM**(Sales) **FROM** SALES-Base **GROUP BY** Region
Q5	**SELECT** 'ALL', 'ALL', 'ALL', **SUM**(Sales) **FROM** SALES-Base

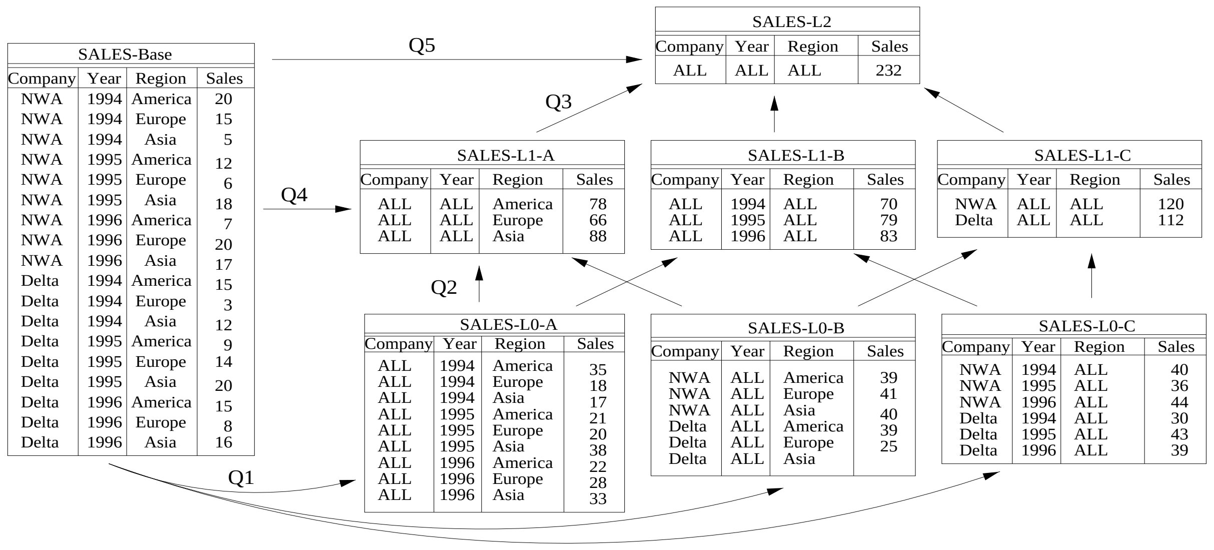
Table 3: Table of **GROUP BY** queries

2.4 What is an Aggregation Hierarchy used for?

To support OLAP, the data cube provides the following operators : roll-up, drill-down, slice and dice, and pivot. We now define these operators.

- Roll-up: increasing the level of abstraction. It generalizes one or more dimensions and aggregate the corresponding measures.

  Table SALES-L0-A in Figure 7 is the roll-up of Table SALES-Base on the Company dimension.



SALES-Base

Company	Year	Region	Sales
NWA	1994	America	20
NWA	1994	Europe	15
NWA	1994	Asia	5
NWA	1995	America	12
NWA	1995	Europe	6
NWA	1995	Asia	18
NWA	1996	America	7
NWA	1996	Europe	20
NWA	1996	Asia	17
Delta	1994	America	15
Delta	1994	Europe	3
Delta	1994	Asia	12
Delta	1995	America	9
Delta	1995	Europe	14
Delta	1995	Asia	20
Delta	1996	America	15
Delta	1996	Europe	8
Delta	1996	Asia	16

SALES-L2

Company	Year	Region	Sales
ALL	ALL	ALL	232

SALES-L1-A

Company	Year	Region	Sales
ALL	ALL	America	78
ALL	ALL	Europe	66
ALL	ALL	Asia	88

SALES-L1-B

Company	Year	Region	Sales
ALL	1994	ALL	70
ALL	1995	ALL	79
ALL	1996	ALL	83

SALES-L1-C

Company	Year	Region	Sales
NWA	ALL	ALL	120
Delta	ALL	ALL	112

SALES-L0-A

Company	Year	Region	Sales
ALL	1994	America	35
ALL	1994	Europe	18
ALL	1994	Asia	17
ALL	1995	America	21
ALL	1995	Europe	20
ALL	1995	Asia	38
ALL	1996	America	22
ALL	1996	Europe	28
ALL	1996	Asia	33

SALES-L0-B

Company	Year	Region	Sales
NWA	ALL	America	39
NWA	ALL	Europe	41
NWA	ALL	Asia	40
Delta	ALL	America	39
Delta	ALL	Europe	25
Delta	ALL	Asia	

SALES-L0-C

Company	Year	Region	Sales
NWA	1994	ALL	40
NWA	1995	ALL	36
NWA	1996	ALL	44
Delta	1994	ALL	30
Delta	1995	ALL	43
Delta	1996	ALL	39

Figure 7: An example of group-by

- Drill-down: decreasing the level of abstraction or increasing detail. It specializes one or a few dimensions and presents low-level aggregations

  Table SALES-L0-A in Figure 7 is the drill-down of Table SALES-L1-A on the Year dimension.

- Slice and Dice: selection and projection. Slicing into one dimension is very much like drilling one level down into that dimension, but the number of entries displayed is limited to that specified in the slice command. A dice operation is like a slice on more than one dimension. On a 2-dimensional display, for example, dicing means slicing on both the row and column dimensions

Company	Year	Region	Sales
ALL	ALL	America	78

Table 4: Slice on the value America of the Region dimension

Table 4 shows the result of slicing into the value of "America" on the Year dimension from the table SALES-L2 in Figure 7. Slicing into one dimension is very much like drilling one level down into that dimension.

Company	Year	Region	Sales
ALL	1994	America	35

Table 5: Dice on value 1994 of Year dimension and value America of Region dimension

Table 5 shows the result of dicing into the value of "1994" on Year dimension and the value of "America" on Region dimension from table SALES-L2 in Figure 7. A dice operation is like slice on more than one dimension.

- Pivoting: re-orienting the multidimensional view of data. It presents the measures in different cross-tabular layouts

3 Map Cube

In this section, we define the map cube operator, which provides an album of maps to browse results of aggregation. We use a simple example to show the distinction between a data cube and a map cube. We also provide the grammar and the translation rules for the map cube operator.

3.1 Definition

We extend the concept of data cube to spatial domain by proposing the "map cube" operator as shown in Figure 8. Map cube is defined as an operator which takes the input parameters, i.e., Base map, Base table, Cartographic preference, etc., and generates an album of maps for analysis and comparison. It is built from the requirements of spatial data warehouse, that is, to aggregate data across many dimensions looking for trends or unusual pattern related to spatial attributes. The basis of map cube is the hierarchy lattice, either dimension power-set hierarchy, or concept hierarchy, or the mixture of both. In this paper, we will focus on the dimension power-set hierarchy. Figure 9 shows an example of dimension power-set hierarchy. This example has three attributes: **Maker(M)**, **Type(T)**, and **Dealer(D)**. There are eight possible groupings of all attributes. Each node in the lattice corresponds to one group-by. Figure 10 shows a three-dimension concept hierarchy. The car maker can be classified as American car, European car, or Japanese car. American car includes Ford, GM, and Chrysler. For the car type dimension, it can be separated as Sedan and Pickup Trucks. The Sedan can go down one detail level as Small, Medium, Large, and Luxury.

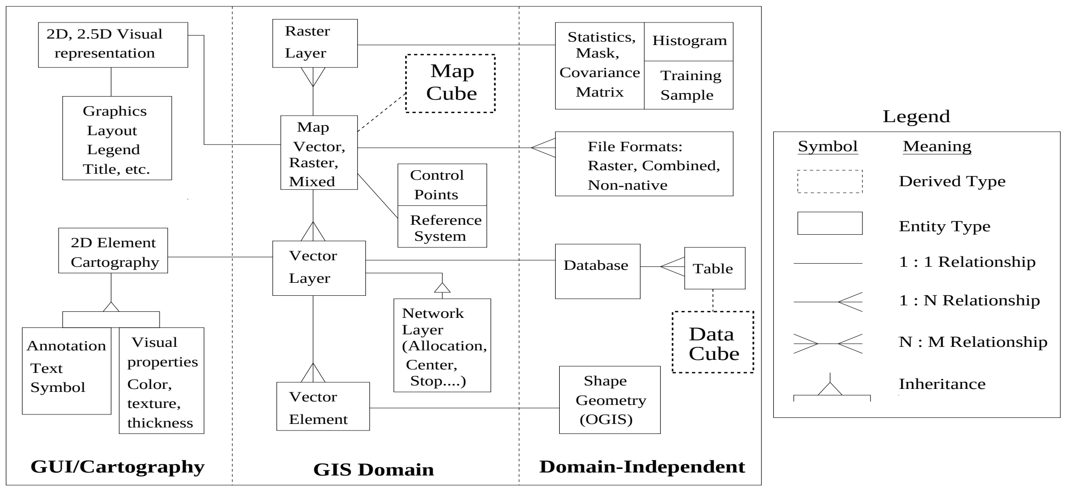


Figure 8: Concepts in GIS with data cube & map-cube

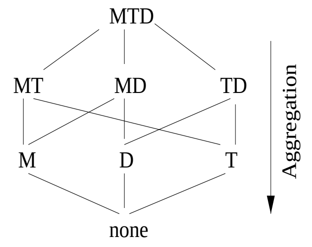


Figure 9: The Dimension Power-Set Hierarchy

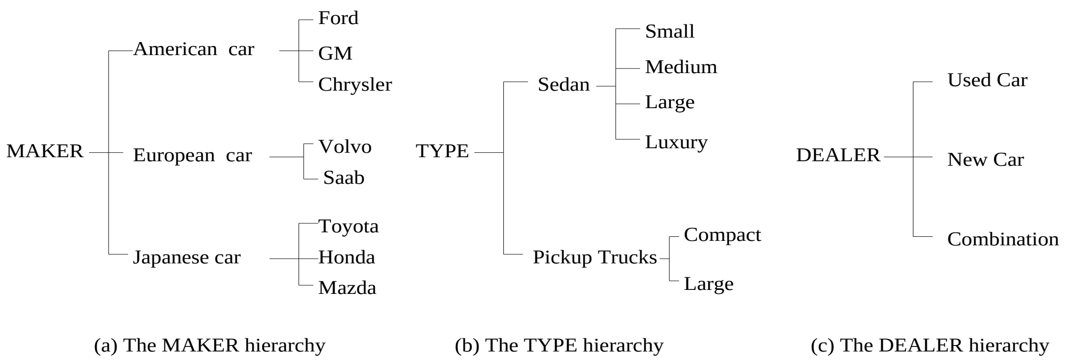


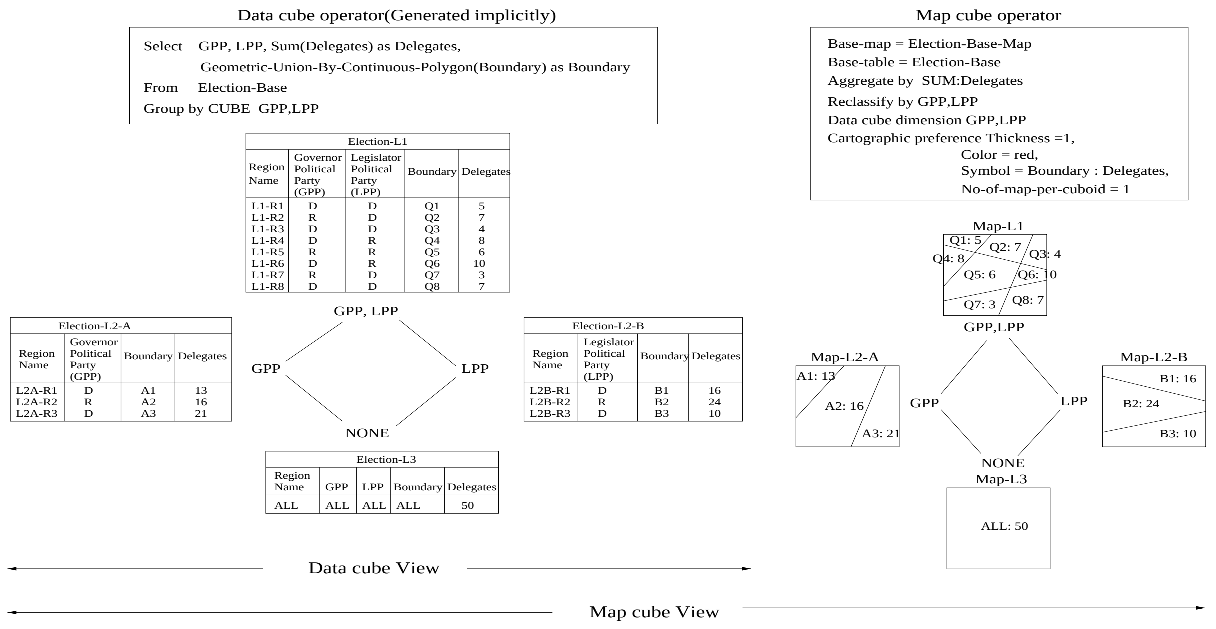
Figure 10: The Concept Hierarchies of (a) MAKER and (b)VEHICLE (c)DEALER

Let the number of dimensions be m, each dimension be A_i, i = 1,2, ... ,m, and A_m be the geographic location dimension, then we have (m-1) different levels of lattice for the dimension power-set hierarchy. Let the level with only one dimension A_i, i = 1,2, ... , m-1, be the 1th level, the level with two dimensions, A_{ij}, where i≠j, i=1,2,...m-1, j=1,2,...m-1, be the 2st level, and the level with the complete set of dimensions $A_1A_2A_3, \ldots, A_{m-1}$ be the (m-1)th level. The total number of cuboids is $\sum_{i=1}^{m-1}\binom{m-1}{i}$. Let the cardinality of each dimension A_i be c_i, then for each cuboid, such as, $A_iA_j...A_k$, we have an album of $c_i \times c_j \times ... \times c_k$ maps.

In other words, map cube is a data cube with cartographic visualization of each dimension to generate an album of related maps for dimension power-set hierarchy or concept hierarchy or mixed hierarchy. Map cube adds more capability to traditional GIS where maps are often independent [2]. The data-cube capability of roll-up, drill-down, slicing and dicing get combined with the map view. It can benefit analysis and decision making based on spatial data warehouses.

3.2 An example of map cube

Figure 11 demonstrates the comparison between a map cube and a data cube. The operators for data cube and map cube are also provided. The left portion of the figure is the effect of data cube operation. There are 2 attributes in the Group by Cube clause, so the data cube will generate 4 tables. The map cube operator not only generates the tables, but also produces the associated map for each table. In the map cube operator, the attributes for "Reclassify by" and "Data cube dimension" are the same, and the parameter, No-of-map-per-cuboid, in the Cartographic preference is set to one, so there is only one map associated with each reclassified table. In other situations, however, each reclassified table is a cuboid, and will generate more than one map after the map cube operation, depending on the number of attributes in the "Data cube dimension." The queries for the data cube and the map cube are also provided at the top of the figure.


Data cube operator(Generated implicitly)

```
Select    GPP, LPP, Sum(Delegates) as Delegates,
          Geometric-Union-By-Continuous-Polygon(Boundary) as Boundary
From      Election-Base
Group by CUBE  GPP,LPP
```

Map cube operator

```
Base-map = Election-Base-Map
Base-table = Election-Base
Aggregate by  SUM:Delegates
Reclassify by GPP,LPP
Data cube dimension GPP,LPP
Cartographic preference Thickness =1,
                        Color = red,
                        Symbol = Boundary : Delegates,
                        No-of-map-per-cuboid = 1
```

Election-L1

Region Name	Governor Political Party (GPP)	Legislator Political Party (LPP)	Boundary	Delegates
L1-R1	D	D	Q1	5
L1-R2	R	D	Q2	7
L1-R3	D	D	Q3	4
L1-R4	D	R	Q4	8
L1-R5	R	R	Q5	6
L1-R6	D	R	Q6	10
L1-R7	R	D	Q7	3
L1-R8	D	D	Q8	7

Election-L2-A

Region Name	Governor Political Party (GPP)	Boundary	Delegates
L2A-R1	D	A1	13
L2A-R2	R	A2	16
L2A-R3	D	A3	21

Election-L2-B

Region Name	Legislator Political Party (LPP)	Boundary	Delegates
L2B-R1	D	B1	16
L2B-R2	R	B2	24
L2B-R3	D	B3	10

Election-L3

Region Name	GPP	LPP	Boundary	Delegates
ALL	ALL	ALL	ALL	50

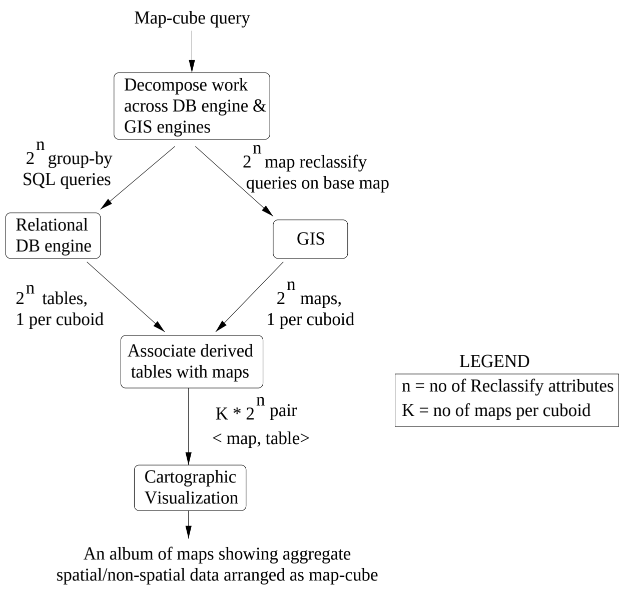
Figure 11: Map cube view

3.3 Steps in Generating Map Cube

To generate the map cube, first, we issue the map-cube query. This query is decomposed into the DB engine and the geometry engine. The relational DB engine processes the 2^n group-by SQL queries and generates 2^n tables, where n is the number of attributes in the "Reclassify by" clause. The GIS procedure processes the map reclassify queries on the base map and generates 2^n maps. These maps and tables are in one-one correspondence with each other. Finally, the cartographic preferences are dealt with in the Cartographic Visualization Step, and an album of maps is plot.

3.4 The Grammar for Map Cube Operator

We have used "yacc" [14] like syntax to describe the syntax of map cube via a grammar and translation rules. Words in angular brackets, e.g. <*carto-value*> , denote non-terminating elements of the language. For example, the <*carto-value*> will be translated into either <*name*> or <*num*>. The vertical bar(|) means "or," and the parentheses are used to group subexpressions. The star(*) denotes zero or more instances of the expression, and the plus(+) denotes one



Map-cube query
Decompose work across DB engine & GIS engines
2^n group-by SQL queries
2^n map reclassify queries on base map
Relational DB engine
GIS
2^n tables, 1 per cuboid
2^n maps, 1 per cuboid
Associate derived tables with maps
K * 2^n pair < map, table>
Cartographic Visualization
An album of maps showing aggregate spatial/non-spatial data arranged as map-cube
LEGEND
n = no of Reclassify attributes
K = no of maps per cuboid

Figure 12: Steps in Generating Map Cube

or more instances of the expression. The unary postfix operator(?) means zero or one instance of the expression. These translations will continue till we reach a terminating elements. For example, <*letter*> will finally be translated into one character.

An example of this grammar is shown in figure 11. In this map cube operator, the following translations are done: <*base-map name*> into Election-Base-Map, <*base-table name*> into Election-Base, <*aggregate list*> into {SUM:Delegates}, <*attribute list*> of both "Reclassify by" and "Data cube dimension" into {GPP,LPP}, and <*carto attribute list*> into {Thickness=1,Color=red,Symbol=Boundary:Delegates,No-of-map-per-cuboid=1}.

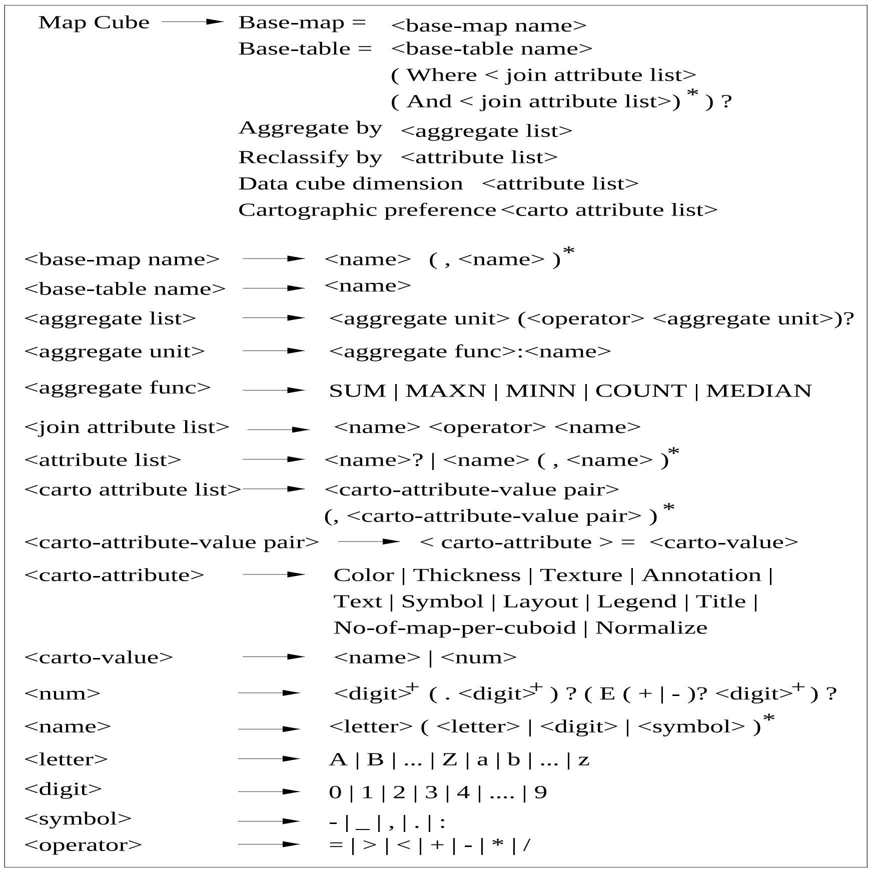
4 A Case Study - Analyzing the Census Data

4.1 Application Domain

The 1990 Census is the most detailed tabulation of Americans demographic data. It contains detailed data on population, race and ethnicity, age and sex, education, employment, income, poverty, housing, and many other topics for each of several different levels of geography:

- The United States and major regions of the country
- Each state and metropolitan area
- All 3,000+ counties in the United States
- Municipalities, census tracts, and block groups

In this case study, we summarize data for the seven-county Twin Cities metropolitan area: Hennepin, Ramsey, Anoka, Carver, Dakota, Scott, and Washington Counties. We use census tracts as the unit of analysis, since census tract provides better geographic detail than city-level



```
Map Cube ——► Base-map =  <base-map name>
             Base-table = <base-table name>
                          ( Where < join attribute list>
                          ( And < join attribute list>) * ) ?
             Aggregate by  <aggregate list>
             Reclassify by  <attribute list>
             Data cube dimension  <attribute list>
             Cartographic preference <carto attribute list>

<base-map name>       ——►  <name>  ( , <name> )*
<base-table name>     ——►  <name>
<aggregate list>      ——►  <aggregate unit> (<operator> <aggregate unit>)?
<aggregate unit>      ——►  <aggregate func>:<name>
<aggregate func>      ——►  SUM | MAXN | MINN | COUNT | MEDIAN
<join attribute list> ——►  <name> <operator> <name>
<attribute list>      ——►  <name>? | <name> ( , <name> )*
<carto attribute list>——►  <carto-attribute-value pair>
                           (, <carto-attribute-value pair> ) *
<carto-attribute-value pair>  ——►  < carto-attribute > = <carto-value>
<carto-attribute>     ——►  Color | Thickness | Texture | Annotation |
                           Text | Symbol | Layout | Legend | Title |
                           No-of-map-per-cuboid | Normalize
<carto-value>         ——►  <name> | <num>
<num>                 ——►  <digit>+ ( . <digit>+ ) ? ( E ( + | - )? <digit>+ ) ?
<name>                ——►  <letter> ( <letter> | <digit> | <symbol> )*
<letter>              ——►  A | B | ... | Z | a | b | ... | z
<digit>               ——►  0 | 1 | 2 | 3 | 4 | .... | 9
<symbol>              ——►  - | _ | , | . | :
<operator>            ——►  = | > | < | + | - | * | /
```

Figure 13: The Grammar for map cube operator

data. Census tracts contain an average of 4,000 people. In cities like Minneapolis or Saint Paul, a census tract is often ten-to-twenty city blocks in size. Minneapolis contains 126 census tracts and Saint Paul, 82.

4.2 Map cube definition

4.2.1 Base Table

In this case study, there are four dimensions and one measure. The four dimensions are county location, age group, income type and race category. The measure is population. The location dimension is embedded in the map, and we should consider the other three dimensions in the power-set hierarchy(see Figure 14). We now provide a detailed explanation for each dimension:

- Age Group: There are seven groups, under 25 years, 25 to 34 years, 35 to 44 years, 45 to 54 years, 55 to 64 years, 65 to 74 years, 75 year and over;
- Income type: There are nine types, less than \$5,000, \$5,000 to \$9,999, \$10,000 to \$14,999, \$15,000 to \$24,999, \$25,000 to \$34,999, \$35,000 to \$49,999, \$50,000 to \$74,999, \$75,000 to \$99,999, \$100,000 or more;

- Race category: There are six categories, (white), (black), (American Indian, Eskimo, or Aleut), (Asian or pacific islander), (other);

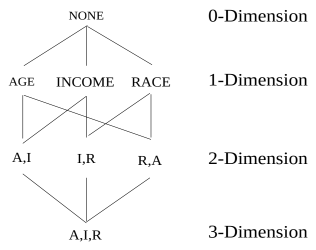


Figure 14: Census data dimension power-set hierarchy

Table 6 and table 7 are two examples of the fundamental tables for constructing the map cube. Table 6 describes the spatial attributes and spatial concept hierarchy for each census track. The Boundary attribute contains the latitude and longitude of each point in the census track boundary. Table 7 contains the non-spatial attributes (Age Group, Income Type, Race category) and measure (number of people) for each census track.

CT-B			
Census-Track-ID	Boundary	County	State
10123	(,), (,),...	Dakota	Minnesota
10186	(,), (,),...	Hennepin	Minnesota

Table 6: Base Table CT-B for Census Data

Population(Householder)				
Census-Track-ID	Age	Income	Race	No-of-people(House Holder)
10123	35-44	\$5,000 to \$9,999	White	40
10123	35-44	\$10,000 to \$14,999	White	25

Table 7: Base Table Population for Census Data

4.2.2 Map cube 0-D SQL query

In this dimension, there is one map corresponding to householder population, which is the summary of all the age group, income type, and race category. The householder population density is defined as the the number of householders divided by the size of the region. The householder population density data are listed in table 8. In this table, the householder population density is convert to the range of 0 to 1, which is the gray scale range. The query is to generate the map and reclassify the original map from a census block unit to be a county level block.

The query for the 0-D map cube:

Base-map=CT-B-Map
Base-table=CT-B,Population
Where CT-B.Census-Track-ID = Population.Census-Track-ID
Aggregate by Sum:No-of-people / Sum:Area(CT-B.boundary)
Reclassify by CT-B.County
Data cube dimension
Cartographic preference Color=grayscale,No-of-map-per-cuboid=one per category

The query for the 0-D data cube:

Select Geometric-union(CT-B.Boundary), Sum(No-of-people)/Sum(Area(CT-B.boundary))
From CT-B,Population
Where CT-B.Census-Track-ID = Population.Census-Track-ID
Group by CT-B.County

County Name	Householder population Density (Householders / Sq km)	Gray Scale
Anoka	75.29	0.156
Carver	18.03	0.037
Dakota	67.39	0.140
Hennepin	293.0	0.610
Ramsey	480.3	1.000
Scott	21.07	0.043
Washington	48.88	0.101

Table 8: Householder population density in seven counties of Twin Cities

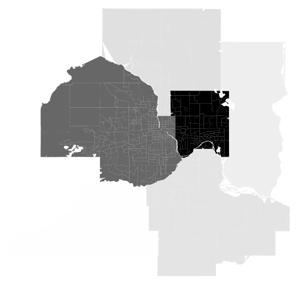


Figure 15: House-holder population density in seven counties of Twin Cities

Table 8 and Figure 15 show the map cube and the associated data cube generated from the above map cube query. This 0-dimensional map cube allows users to observe the relative householder population densities of the seven counties within the Twin Cities metropolitan area. The data used based on 1990 census data. The maps are for illustrative purposes only since data quality was not evaluated. Located in the center area, the Ramsey county has the highest householder population density with the dark region, while the Carver county, located in the Southwest of the map, has the lowest householder population density, as we can easily observe from the map.

4.2.3 Map cube 1-D SQL query

In this dimension, there are seven maps, one for each age group. This query reclassifies the base map from a census block unit to a county level block, and generates corresponding map for each age group.

The query for the 1-D map cube:

Base-map=CT-B-Map
Base-table=CT-B,Population
Where CT-B.Census-Track-ID = Population.Census-Track-ID
Aggregate by Sum:No-of-people
Reclassify by CT-B.County
Data cube dimension Age
Cartographic preference Color=grayscale,No-of-map-per-cuboid=one per category,
Normalize=CT-B.County(Population)

The query for the 1-D data cube:

Select Geometric-union(CT-B.Boundary), Sum(No-of-people)
From CT-B,Population
Where CT-B.Census-Track-ID = Population.Census-Track-ID
Group by CT-B.County, Age

Table 9 and Figures 16, 17 show the map cube and the associated data cube generated from the above map cube query. This 1-dimensional map cube allows the users to observe the fraction of the age group within each county. Note that the householder population is normalized within each county, which is specified by the cartographic preferences. A few interesting trends can be observed from the map cube. Comparing age-groups shows that most counties have more householders with middle-age , i.e. age groups 25-34, 35-44, 45-54, than other age groups. Comparing counties shows that the central counties, Ramsey and Hennepin, have higher fraction of seniors as well as children.

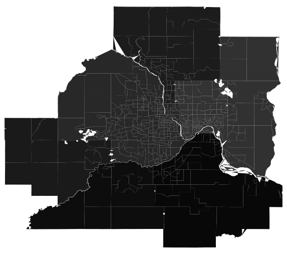
4.2.4 Map cube 2-D SQL query

In this dimension, there are sixty-three maps corresponding to the combination of the two dimensions (seven age groups and nine income types). This query reclassifies the base map

County Name	Age <25	Age 25-34	Age 35-44	Age 45-54	Age 55-64	Age 65-74	Age >75	Total
Anoka	0.046	0.278	0.275	0.183	0.111	0.069	0.035	1
Carver	0.041	0.280	0.253	0.158	0.112	0.079	0.074	1
Dakota	0.052	0.300	0.273	0.163	0.101	0.066	0.043	1
Hennepin	0.063	0.265	0.235	0.142	0.112	0.099	0.080	1
Ramsey	0.064	0.252	0.223	0.135	0.114	0.108	0.092	1
Scott	0.038	0.295	0.270	0.165	0.097	0.073	0.059	1
Washington	0.027	0.254	0.285	0.198	0.117	0.070	0.046	1

Table 9: Fraction of people in the same age group within each county, normalized within each county

(a) Fraction of age group <25


(b) Fraction of age group 25-34

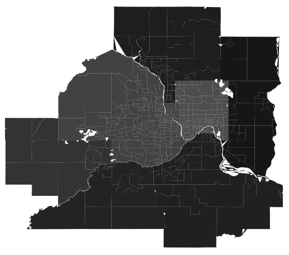

(c) Fraction of age group 35-44

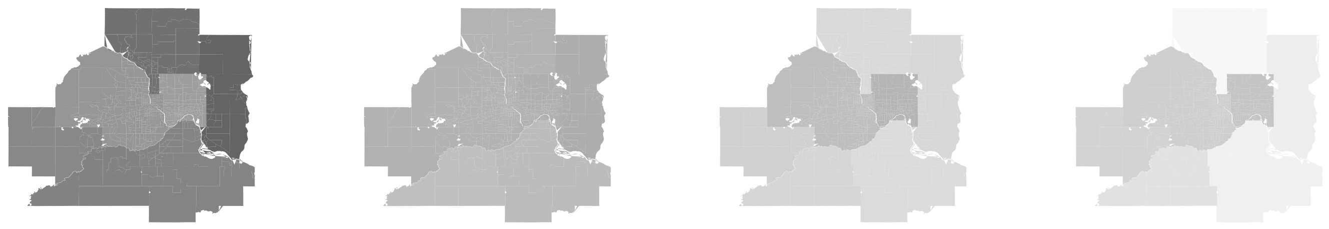
Figure 16: Fraction of people in the same age group within each county

from a census block unit to a county level block, and generates corresponding map for each combination of age group and income level.

The query for the 2-D map cube:

Base-map=CT-B-Map
Base-table=CT-B,Population
Where CT-B.Census-Track-ID = Population.Census-Track-ID
Aggregate by Sum:No-of-people
Reclassify by CT-B.County
Data cube dimension Age, Income
Cartographic preference Color=grayscale,No-of-map-per-cuboid=one per category,
Normalize=CT-B.County(Population)

The query for the 2-D data cube:



(a) Fraction of age group 45-54

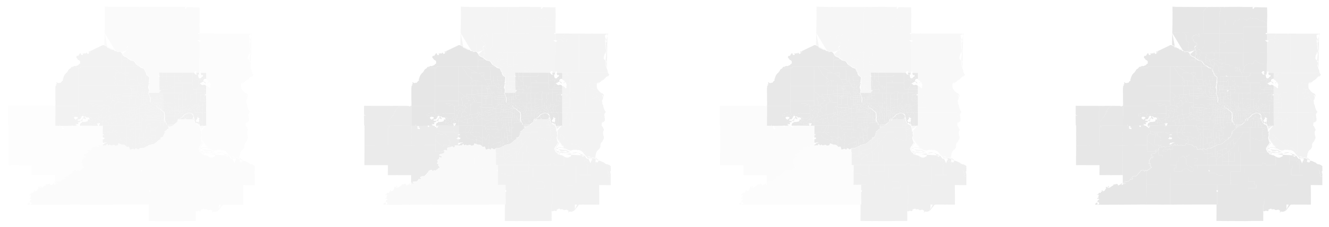
(b) Fraction of age group 55-64

(c) Fraction of age group 65-74

(d) Fraction of age group >75

Figure 17: Fraction of people in the same age group within each county

Select Geometric-union(CT-B.Boundary), Sum(No-of-people)
From CT-B,Population
Where CT-B.Census-Track-ID = Population.Census-Track-ID
Group by CT-B.County, Age, Income



(a) Fraction of age group <25 and income level <\$5,000

(b) Fraction of age group <25 and income level \$5,000-\$9,999

(c) Fraction of age group <25 and income level \$10,000-\$14,999

(d) Fraction of age group <25 and income level \$15,000-\$24,999

Figure 18: Fraction of the age group <25, different income levels

Table 10 and Figures 18, 19 show the map cube and the associated data cube generated from the above map cube query. This 2-dimensional map cube allows the users to observe the fraction of age group and income level within each county, where the householder population is normalized within each county. In the figures, we only display the maps with age group less than 25. Within this age group, there are nine maps associated with these nine income levels. There are few householders with age less than 25 and have income more than \$100,000. Therefore, in the Figure 19(e), we see a nearly empty map.

County Name	Age Group	less than $5,000	$5,000 - $9,999	$10,000 - $14,999	$15,000 - $24,999	$25,000 - $34,999	$35,000 - $49,999	$50,000 - $74,999	$75,000 - $99,999	$100,000 or more	County Total
Anoka	<25	0.002	0.004	0.003	0.012	0.010	0.008	0.003	0.000	0.000	1
	25-34	0.004	0.009	0.009	0.036	0.054	0.096	0.056	0.007	0.003	
	35-44	0.002	0.004	0.006	0.026	0.045	0.080	0.080	0.019	0.009	
	45-54	0.001	0.001	0.003	0.013	0.023	0.046	0.061	0.021	0.009	
	55-64	0.002	0.004	0.004	0.014	0.018	0.027	0.025	0.009	0.004	
	65-74	0.002	0.008	0.010	0.017	0.012	0.009	0.005	0.000	0.000	
	>75	0.003	0.010	0.006	0.007	0.003	0.002	0.000	3.613	0.004	
Carver	<25	0.000	0.007	0.001	0.009	0.008	0.008	0.003	0.000	0.000	1
	25-34	0.002	0.007	0.009	0.034	0.057	0.085	0.060	0.014	0.007	
	35-44	0.002	0.003	0.007	0.018	0.042	0.069	0.067	0.024	0.014	
	45-54	0.001	0.002	0.003	0.017	0.021	0.028	0.042	0.021	0.017	
	55-64	0.002	0.004	0.006	0.013	0.017	0.022	0.027	0.007	0.009	
	65-74	0.004	0.010	0.013	0.019	0.011	0.009	0.007	0.001	0.001	
	>75	0.008	0.022	0.012	0.016	0.008	0.003	0.001	0.000	0.005	
Dakota	<25	0.001	0.005	0.006	0.012	0.010	0.010	0.004	0.000	0.000	1
	25-34	0.003	0.008	0.007	0.035	0.058	0.093	0.073	0.014	0.005	
	35-44	0.002	0.004	0.005	0.022	0.038	0.069	0.083	0.028	0.018	
	45-54	0.001	0.001	0.003	0.011	0.016	0.032	0.056	0.023	0.016	
	55-64	0.002	0.002	0.003	0.014	0.015	0.018	0.024	0.009	0.009	
	65-74	0.002	0.006	0.008	0.016	0.012	0.009	0.005	0.001	0.001	
	>75	0.004	0.010	0.007	0.010	0.004	0.003	0.001	0.000	0.003	
Hennepin	<25	0.005	0.010	0.008	0.016	0.011	0.008	0.002	0.000	0.000	1
	25-34	0.007	0.015	0.014	0.042	0.047	0.067	0.050	0.011	0.006	
	35-44	0.004	0.008	0.009	0.027	0.034	0.053	0.056	0.021	0.020	
	45-54	0.003	0.004	0.004	0.013	0.016	0.027	0.036	0.017	0.018	
	55-64	0.004	0.005	0.005	0.015	0.015	0.022	0.023	0.009	0.011	
	65-74	0.003	0.011	0.012	0.023	0.018	0.014	0.009	0.002	0.003	
	>75	0.005	0.020	0.013	0.018	0.008	0.006	0.004	0.000	0.004	
Ramsey	<25	0.005	0.011	0.009	0.016	0.010	0.008	0.002	0.000	0.000	1
	25-34	0.007	0.016	0.016	0.051	0.049	0.058	0.040	0.007	0.003	
	35-44	0.005	0.010	0.009	0.029	0.035	0.053	0.053	0.015	0.011	
	45-54	0.003	0.004	0.005	0.013	0.018	0.028	0.033	0.016	0.013	
	55-64	0.004	0.007	0.007	0.016	0.017	0.023	0.022	0.009	0.007	
	65-74	0.005	0.014	0.014	0.026	0.017	0.017	0.009	0.002	0.002	
	>75	0.006	0.026	0.016	0.019	0.010	0.006	0.004	0.001	0.003	
Scott	<25	0.001	0.002	0.001	0.010	0.006	0.011	0.003	0.000	0.000	1
	25-34	0.002	0.006	0.008	0.031	0.054	0.103	0.067	0.013	0.006	
	35-44	0.002	0.004	0.004	0.022	0.042	0.072	0.083	0.022	0.014	
	45-54	0.002	0.002	0.004	0.012	0.017	0.041	0.050	0.017	0.015	
	55-64	0.001	0.001	0.004	0.015	0.019	0.023	0.018	0.008	0.003	
	65-74	0.002	0.011	0.010	0.020	0.010	0.008	0.005	0.000	0.002	
	>75	0.007	0.021	0.009	0.013	0.004	0.002	0.000	0	0.004	
Washington	<25	0.001	0.004	0.002	0.005	0.005	0.005	0.001	0.000	0.000	1
	25-34	0.002	0.008	0.007	0.029	0.048	0.077	0.061	0.012	0.005	
	35-44	0.002	0.004	0.006	0.018	0.033	0.074	0.093	0.028	0.021	
	45-54	0.001	0.002	0.004	0.011	0.018	0.041	0.064	0.031	0.022	
	55-64	0.002	0.003	0.003	0.014	0.015	0.024	0.026	0.014	0.010	
	65-74	0.001	0.011	0.008	0.016	0.012	0.009	0.007	0.001	0.001	
	>75	0.003	0.011	0.009	0.009	0.004	0.004	0.001	0.000	0.004	

Table 10: Fraction of age group and income level within each county, normalized within each county

5 Research Issues

In this section we elaborate on the research issues involving constructing spatial data warehouse, and in particular map cube. The map cube inherits ideas from three different domains, namely, data warehouse, visualization and GIS [4, 21, 13]. A data warehouse is defined as "a subject-oriented, integrated, time-varying, non-volatile collection of data that is used primarily in organizational decision making." A spatial data warehouse is an extension to support spatial databases. Visualization is essentially a mapping process from computer representations to perceptual representations, choosing encoding techniques to maximize human understanding and communication. The primary objective of data visualization is to gain insight into an information space by mapping data onto graphical primitives [19]. A map is a graphic representation of spatial relationships and spatial forms. Cartography is the art, science and technology of making maps [15]. A GIS is best defined as a system which uses a spatial database to provide

(a) Fraction of age group <25 and income level $25,000-$34,999

(b) Fraction of age group <25 and income level $35,000-$49,999

(c) Fraction of age group <25 and income level $50,000-$74,999

(d) Fraction of age group <25 and income level $75,000-$99,999

(e) Fraction of a group <25 a income level $100,000

Figure 19: Fraction of the age group <25, different income levels

answers to queries of geographical nature [7]. The three-way interaction model is shown in figure 20. The tertiary intersection is our topic “map cube.”

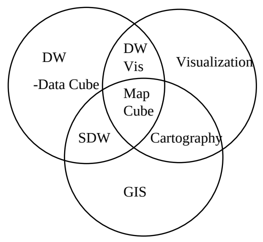


Figure 20: The relationship of Map Cube with three parent domains

5.1 Data Warehouse

5.1.1 Distinction between dimensions and attributes

To build a data cube from the original dataset, not all the attributes in the table can be represented as dimensions in the cube. Here, we define the concept of full functional dependency. A functional dependency $X \rightarrow Y$ is a full functional dependency if the removal of any attribute A from X means that the dependency does not hold any more. For example, in figure 21, we have three sets of full functional dependency, {Model,Year,Region}→{Profits}, {Model,Year,Region}→{# of Customers}, and {Model,Year,Region}→{Sales}. The attributes Model, Year, and Region can be used as the dimensions in the aggregate of Profits, # of customers, and Sales.

5.1.2 Are all nodes in dimension hierarchy meaningful in map cubes?

For geographic data, the nodes in higher dimension may not be meaningful. The reason is two-folded. First, the sample size is low in higher dimension, so the statistical significance

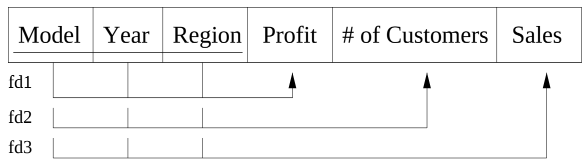


Figure 21: Example of functional dependency

is decreased, such as the mean and median functions. Secondly, the privacy issue, as in the case of census data, only the block or census track data can be revealed, but not the personal information. However, this constraint could be violated in high dimensions of the map cube. We illustrate this with the following example.

County	Census Track	Age	Income	Race	No of people
Dakota	24	35 - 44	\$75,000 to \$99,999	White	2
Dakota	24	35 - 44	\$75,000 to \$99,999	Black	1
Dakota	24	35 - 44	\$75,000 to \$99,999	Asian	2
Dakota	24	35 - 44	\$100,000 or more	White	2
Dakota	24	35 - 44	\$100,000 or more	Black	2
Dakota	24	35 - 44	\$100,000 or more	Asian	1

Table 11: 3D data cube

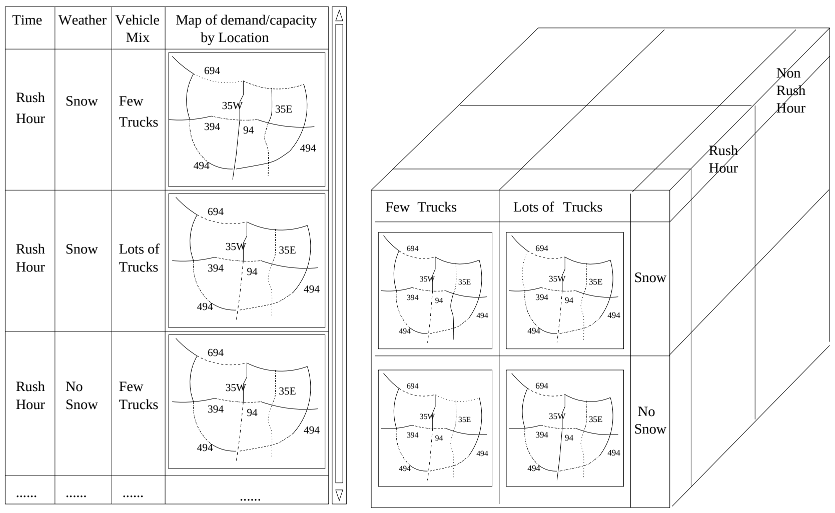
County	Census Track	Age	Income	No of people
Dakota	24	35 - 44	\$75,000 to \$99,999	5
Dakota	24	35 - 44	\$100,000 or more	5

Table 12: 2D data cube

Table 11 shows the data cube of the census track, from which we can easily infer the income of a specific person if there is only one person in a certain category, and this will violate the issue of personal privacy. Table 12 shows the aggregate data of the 2D data cube, which satisfies the personal privacy constraint. So while exploring the census data, the answer to the question, should the map cube generate the full detail cube, is clearly no as it violates the privacy constraint. This brings on another question "To what extent can the users explore?" While this is subject specific, the map cube should provide constraint based exploration mechanism.

5.1.3 Given a cuboid in the dimension hierarchy, is the tabular representation adequate?

Figure 22 shows the tabular and cube views of the traffic data. The tabular view provides a scroll bar for browsing the whole combination of the categories, including the ALL function in each dimension. The data can be sorted on each dimension, and the users can perform the Select, Project, and Join operations. In contrast, the cube view of the traffic data provides additional operations, such as slice or dice. Given the dimension lattice, the user will browse



(a) The tabular view of the data (b) The cube view of the data

Figure 22: Tabular view and cube view

the subcube by using the slice or dice operation. Which representation is more suitable for representing the map cube and more convenient for users to browse the requested information, tabular or cube view?

5.1.4 How can the Map Cube be efficiently implemented?

A map cube is an extension of a conventional data cube, which utilizes cartographic techniques for efficient exploration of spatial data. The conventional data cube operations are designed to work with the data stored in relational DBMS, which deals with few well-defined data types, but not with geographic data types. Therefore, new algorithms are needed when dealing with spatial data in order to provide the tabular or pivot view of the map cube. Furthermore, a map cube and an ordinary data cube require different implementation algorithms.

Selection of Groupbys for Precomputation

Each cell of the map cube is a view consisting of an aggregation of interest. The values of many of these cells are dependent on the values of other cells in the map cube. A query optimization technique is used to materialize some or all of these cells rather than computing them from raw data cubes. The important objective is to develop techniques for optimizing the space-time tradeoff when implementing a lattice of views, which is known to be an NP-complete problem. Harinarayan, et al [10] investigated the issue of which cells(views) to materialize when it is too expensive to materialize all views. Gupta [9] proposed a framework a framework for the selection of views to optimize the query response time within polynomial time.

The analysis of whether a cuboid should be selected for materalization in a spatial data cube is similar to that in a non-spatial one. The spatial computation, such as region merging, map overlaying, spatial join, could be expensive, especially when a large number of spatial objects are involved. There is an additional factor to be considered for a spatial data cube: the cost of on-line computation of a particular spatial cuboid. Han, et al [11] proposed some techniques for selective materalization of the spatial computation results. They assumed that a set of cuboids have been selected for materalization using an extended cuboid-selection algorithm similar to the one in [10], and examined how to determine which sets of mergeable spatial objects should be precomputed. Two algorithms, *pointer intersection* and *object connection*, are worked out for the selection of precomputed objects.

5.1.5 Is the location one dimension or two or more?

Figure 23 shows how to aggregate the data on a three dimensional Earth space. Initially, the cuboid has three dimensions < Latitude,Longitude,Altitude>. Then the data is aggregated into two dimensions. At the bottom level, all data are summarized into one point.

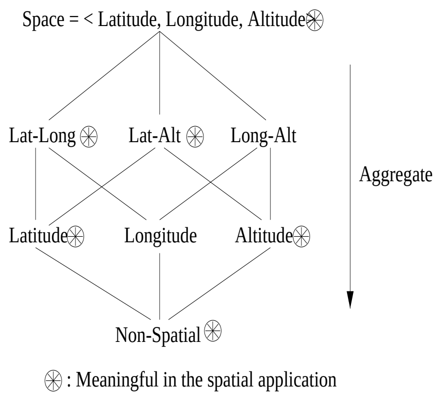

Space = < Latitude, Longitude, Altitude ⊛
Lat-Long ⊛
Lat-Alt ⊛
Long-Alt
Aggregate
Latitude ⊛
Longitude
Altitude ⊛
Non-Spatial ⊛
⊛ : Meaningful in the spatial application

Figure 23: The space lattice

The cuboids in the lattice that are meaningful for certain applications, e.g. weather analysis, are marked. For example, groupby Altitude is important for analyzing weather patterns in each Elevation. However, it makes no sense to group by Longitude. In designing a map cube, we can either allow the users to explore all subspaces(cuboids) of the three dimensional geographic space, or we may restrict subspaces(cuboids) to meaningful ones.

Choice	Allow all sub-space	Restrict to useful subspace
Computation Effort	High	Low, since useless subspaces are not computed
Semantic Integrity	Low, since some maps generated in map-cube may be violating geographic constraints	High, all maps are meaningful

Table 13: Decision to allow all sub-spaces to be computed or not

5.2 Cartographic Visualization

As discussed in the previous sections, a map is the core output of a map cube. A map cube can be defined as a graphic depiction of all or part of geographic realm in which the real-world features have been replaced with symbols in their correct spatial location at a reduced scale. The process of creating graphic symbols to represent feature attribute values is part of what we call symbolization. The symbolization process is carried out after applying classification, simplification and exaggeration processes to the database selected for mapping. Hence we have two separate components of the map design process. The first encodes information, and is an abstract process related to generalization. The second involves conceptual constraints and focuses on graphic representation. Both components have a theoretical basis in the field of semiotics. Semiotics addresses the relations among symbols, signs, and meanings, and forms an appropriate basis for cartographic symbolization. Geographic features have a dimensionality ranging from zero to three. A point feature is dimensionless (zero dimension) while a line feature has one dimension. An area feature has two dimensions and a volume feature has three dimensions.

5.2.1 Issues involved in visualization of geographic features

Geographic features conceived as points (eg. lamp post, building or even city depending on scale) are portrayed by point symbols. On a nominal (qualitative) scale, points can be depicted by symbols with differentiating visual variables (like shape, hue(color) and orientation). Quantitative point attribute data can be symbolized by using primary visual variable size. Geographic features conceived as lines (e.g. rivers, roads, ...) can be portrayed by line symbols. Nominally scaled line features can be depicted by primary visual variables (hue and shape) and quantitative line features are symbolized using visual variables of size and hue. Features conceived as areas (like thematic maps, or remote sensing images) are depicted by secondary visual variables (texture, arrangement, and orientation). Multi-channel images are shown as either black and white (with varying gray shades), or RGB (by assigning different channels to each of red, blue and green color guns). Now several questions arise:

- Is the classification and symbolization process sufficient to visualize complex query maps resulting from aggregation of multi-dimensions?.
- How can a glyph family be designed in a way such that the members can be related in a manner similar to the data cube dimension hierarchy?
- What is the effect of data cube on map legend design and other cartographic issues?
- How to visualize the trends across spatial dimensions, as well as non-spatial dimensions?

In the map cube, not only do the users compare different measures in all regions within one map, but they may also be interested in measuring the variations across different maps. It is not easy to show both the trends within a map and across different maps at the same level.

For example, Table 14 shows the population ratio for each age group in each county. The views that users may be interested in are the variation across different counties within that same age group, and the variation across different age groups in the same county. The most suitable cartographic representation is a map of counties, and within each county, there is a pie-chart for different age groups.

County-name	under 25	25 to 34	35 to 44	45 to 54	55 to 64	65 to 74	75 and over
Anoka	0.045	0.277	0.274	0.183	0.110	0.068	0.039
Carver	0.041	0.279	0.251	0.157	0.111	0.078	0.079
Dakota	0.051	0.299	0.272	0.162	0.101	0.066	0.045
Hennepin	0.063	0.264	0.235	0.142	0.112	0.098	0.082
Ramsey	0.064	0.253	0.225	0.136	0.115	0.109	0.095
Scott	0.037	0.294	0.269	0.164	0.096	0.073	0.063
Washington	0.027	0.254	0.284	0.197	0.116	0.070	0.049

Table 14: Fraction of people in the same age group within each county

5.3 Geographic Information System

5.3.1 Which application can benefit from a map cube?

In our previous examples, we have shown the usefulness of map cube in analyzing traffic and census data. In summary, a map cube can be applied to any spatial application which needs comparisons between different attributes and requires some aggregation functions within each attribute.

5.3.2 How to integrate new aggregate functions into a map cube?

Aggregate operations are domain specific. Thus, the aggregate operations in a conventional data cube, which can be realized by a groupby, may not be directly applicable to spatial databases. For example, the arithmetic addition of two images may not give a meaningful result, whereas the same operation on an attribute (say salary) in a database will be valid. Therefore, it is necessary to find and integrate suitable operators for spatial databases. In this section we discuss some of these issues.

Applied Spatial Statistics: Applied spatial statistics deals with those methods that are applicable to the data that are spatially correlated. Some of the methods which are interesting to spatial data warehouses include moving window statistics (e.g. calculating mean, variance etc,. within a small moving window), spatial clustering, cross-correlation, cross-covariance, spatial autocorrelation and spatial sampling methods. Examples where these methods are needed include calculating the number of neighbors within a distance D, and grouping crimes by the nearest police station etc.

Local operation: A local operation acts upon one or more spatial field functions of the field-model to produce a new field. The value of the new field at a location is dependent only on the value of the original field at that location. Examples include pointwise sums, differences in maximums, minimums, or means, etc.

Focal operation: For a focal operation, the attribute value derived at a location x may depend not only on the attributes of the input spatial field function at x, but also on the attributes of these functions in the neighborhood $n(x)$ of x. Examples include slope, aspect, and weighted average of neighborhood.

Zonal operation: A zonal operation aggregates values of a field over each of a set of zones in the framework. Examples include sum, mean, or maximum of field value in each zone.

Geometric Union, Geometric Intersection: Aggregation in spatial dimension includes geometric union and intersection. For example, at a lower dimension we may have a state boundary map and in the next dimension we may have country maps which are derived from state boundary maps by applying geometric union operation(a spatial aggregate function).

NDVI: In the case of satellite images, the normalized density vegetation index, obtained by rationing the difference between near infra-red and red to their sum. This rationed image highlights (or distinguishes) vegetative regions from non vegetative regions.

$$NDVI = \frac{NearInfraredData - VisibleData}{NearInfraredData + VisibleData} \tag{1}$$

The higher values of the NDVI reveal pixels dominated by high proportions of green biomass.

OGIS: Much work has been done over the last decade on the design of spatial Abstract Data Types(ADTs) and their embedding in a query language. Recently, the OGIS [18] consortium has proposed a specification for incorporating 2D geospatial ADTs in SQL. The present OGIS standard is not sufficient to construct complex spatial data warehouses, which include many other data types and operations not covered by OGIS.

5.3.3 Is scale operation the same as concept hierarchy?

The scale operation changes the size of the object in the embedding plane without changing the shape, position or orientation of the object. For example, the scaling of $'a'$ in the x-direction and $'b'$ in the y-direction is effected by the rule:$(x, y) \rightarrow (ax, by)$. To build the concept hierarchy by the scale operation, we plot the fixed size grid in the embedding plane. Initially, all the map objects are within one grid, which is the top level in the hierarchy. As we scale up the map, the map will extend to the surrounding grids, by which we construct the lower level in the hierarchy. The effect of scaling up/down can also be accomplished by changing the size of the grid in the embedding plane.

6 Conclusion and Future Work

The cube operator generalizes and unifies several common and popular concepts: aggregation, group-by, histogram, roll-up, drill-down, and cross-tab. A map is the core of Geographic Information System. In this paper, we extended the concept of data cube to spatial domain via the proposed "map cube" operator, which is built from the requirements of spatial data warehouses. We defined the "map cube" operator, provided the grammar and translation rules, and used the census data as an application of map cube. Since map cube inherits ideas from three different domains, namely, data warehouse, visualization and GIS, we also discussed the research issues involved with these domains. In the future, we would like to explore the implementation and application of map cube in a spatial data warehouse.

7 Acknowledgment

This work is sponsored in part by the Army High Performance Computing Research Center under the auspices of the Department of the Army, Army Research Laboratory cooperative agreement number DAAH04-95-2-0003/contract number DAAH04-95-C-0008, the content of which does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred. This work was also supported in part by NSF grant #9631539.

References

[1] In *URL: http://www.census.gov.*

[2] BEDARD, Y, 1999. Visual modeling of spatial databases towards spatial extensions and uml. In *Geomatica.*

[3] CHAUDHURI, S. and DAYAL, U., 1996. An overview of data warehousing and olap technology. In *Proc. VLDB Conference.*

[4] CHRISMAN, N., 1997. *Exploring Gepgraphic Information Systems.* John Wiley and Sons.

[5] ESRI. Spatial data warehousing. http://www.esriau.com.au/wp.htm.

[6] FERGUSON, P., 1999. Census 2000 behinds the scenes. In *Intelligent Enterprise*, October.

[7] GOODCHILD, M.F., 1985. Geographic information systems in undergraduate geography: A contemporary dilemma. The Operational Geographer.

[8] GRAY, J., BOSWORTH A., LAYMAN, A. and PIRAHESH, H., 1996. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-total. In *Proceedings of the Twelfth IEEE International Conference on Data Engineering.*

[9] GUPTA, H., 1997. Selection of Views to Materialize in a Data Warehouse. In *International Conference on Database Theory, Delphi, Greece*, January.

[10] HAIRNARAYAN, V., RAJARAMAN, A. and ULLMAN, J., 1996. Implementing data cube efficiently. In *Proc. ACM-SIGMOD Int. Conf. Management of Data.*

[11] HAN, J., STEFANOVIC, N. and KOPERSKI, K., 1998. Selective materialization: An efficient method for spatial data cube construction. In *Proc. Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'98).*

[12] KIMBALL, R., REEVES, L., ROSS, M., and THORNTHWAITE, W., 1998. *The Data warehouse Lifecycle Toolkit.* Wiley.

[13] KRAAK, M. J. and ORMELING, F., 1996. *Cartographer: Visualization of Spatial Data.* Longman.

[14] LEVINE, J.R., MASON, T., and BROWN, D., 1992. In *Lex and Yacc.* O'Reilly And Associates.

[15] MEYEN, E (ed.), 1973. Multilingual dictionary of technical terms in cartography. In *International Cartographic Association, Commission II.* Franz Steiner Verlag.

[16] MICROSOFT. Terraserver: A spatial data warehouse. http://www.microsoft.com.

[17] MUMICK, I. S., QUASS, D. and MUMICK, B. S., 1997. Maintenance of data cubes and summary tables in a warehouse. In *SIGMOD.*

[18] OPEN GIS Consortium. OpenGIS Simple Features Specification for SQL. In *URL: http://www.opengis.org/public/abstract.html.*

[19] SENAY, H and IGNATIUS, E., 1994. A knowledge-based system for visualization design. In *IEEE Computergraphics & Applications.*

[20] STONEBRAKER, M., FREW, J., and DOZIER, J., 1993. The sequoia 2000 project. In *Proceedings of the Third International Symposium on Large Spatial Databases.*

[21] TUFTE, E., 1997. *Envisioning Information.* Graphic Press.

[22] USGS. National satellite land remote sensing data archive. In *http://edc.usgs.gov/programs/nslrsda/overview.html.*

[23] WORBOYS, M. F., 1995. *GIS: A Computing Perspective.* Taylor & Francis.